

# LKP Ethernet Card SW 사용 설명서

## Version Information

Tornado 2.0, 2.2

True Leader of Technology



주엘케이일레븐

## 알림

---

여기에 실린 내용은 제품의 성능 향상과 신뢰도의 증대를 위하여 예고없이 변경될 수도 있습니다.

여기에 실린 내용의 일부라도 엘케이일레븐의 사전 허락없이 어떠한 유형의 매체에 복사되거나 저장될 수 없으며 전기적, 기계적, 광학적, 화학적인 어떤 방법으로도 전송될 수 없습니다.

㈜엘케이일레븐

경기도 성남시 중원구 상대원동 191-1 SKn 테크노파크 메가동 1306 호

LKP-400, LKP-401 는 ㈜엘케이일레븐의 등록상표입니다.

## 목 차

알림 .....	2
1. Introduction.....	4
1.1. LKP의 소개.....	4
1.2. 본 문서의 내용 .....	4
2. LKP Ethernet Card .....	5
2.1. IP 설정 방법 .....	5
3. TCP/UDP Program Examples .....	8
3.1. TCP 통신 Server Client Program Examples.....	8

## 그림 목차

그림 1. LKV-080A Board Booting 완료 Image .....	5
그림 2. LKP Card IP 설정 후 ifShow 출력 결과 .....	6

## 1. Introduction

### 1.1. LKP 의 소개

LKP(PCI Mezzanine Card)는 LKV-080A 보드의 Ethernet Port 확장용 Card 이다. LKP 에서 지원되는 Ethernet Port 는 2Port 이다.

### 1.2. 본 문서의 내용

2 장에서는 LKP 의 Setting 방법과 IP 설정 방법에 대해 간단히 설명되며, IP 설정 후 확인 할 수 있는 VxWorks Function 과, TCP/UDP 프로그램의 예를 설명 한다.



[그림 1]과 같이 booting 이 완료된 상태에서 “EnetAttach” 함수를 이용하여 LKP Ethernet Port 의 IP 를 등록 한다. IP 설정 시 각 Port 당 반드시 다른 Network 로 설정 되어야 하며, 사용 예는 다음과 같다.

EX) EnetAttach Function 사용 예

EnetAttach(arg1, arg2, arg3)

Arg1 : Ethernet Device Number

Arg2 : Ethernet IP

Arg3 : NetMask value

```
-> EnetAttach(2,"100.0.0.1",0xffffffff)
value = 0 = 0x0
-> EnetAttach(3,"100.0.1.1",0xffffffff)
value = 0 = 0x0
```



그림 2. LKP Card IP 설정 후 ifShow 출력 결과

[그림 2]와 같이 ifShow 실행 결과로 확인사항은 다음과 같다.

```

fei (unit number 2):
  Flags: (0x8063) UP BROADCAST MULTICAST ARP RUNNING
  Type: ETHERNET_CSMACD
  Internet address: 100.0.0.1
  Broadcast address: 100.0.0.255
  Netmask 0xff000000 Subnetmask 0xfffff00
  Ethernet address is 00:0b:bb:ff:00:15
  Metric is 0
  Maximum Transfer Unit size is 1500
  0 packets received; 1 packets sent
  0 multicast packets received
  0 multicast packets sent
  0 input errors; 0 output errors
  0 collisions; 0 dropped
fei (unit number 3):
  Flags: (0x8063) UP BROADCAST MULTICAST ARP RUNNING
  Type: ETHERNET_CSMACD
  Internet address: 100.0.1.1
  Broadcast address: 100.0.1.255
  Netmask 0xff000000 Subnetmask 0xfffff00
  Ethernet address is 00:0b:bb:ff:00:16
  Metric is 0
  Maximum Transfer Unit size is 1500
  0 packets received; 1 packets sent
  0 multicast packets received
  0 multicast packets sent
  0 input errors; 0 output errors
  0 collisions; 0 dropped
  
```

Unit number 가 각각 순차적으로 설정이 되어야 하며, “Internet address”가 “EnetAttach” Function 을 통해 등록된 Unit Number 와 IP 와 같아야 한다. NetMask 는 IP Class 에 따라 결정되며, A Class, B Class, C Class 에 따라, “0xff000000”, “0xffff0000”, “0xffffffff00”이 된다.

### 3. TCP/UDP Program Examples

#### 3.1. TCP 통신 Server Client Program Examples

VxWorks TCP/UDP Socket 설정 관련 Header File

```
#include <vxWorks.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <types.h>
#include <socket.h>

#include "in.h"
#include "inetLib.h"
#include "sockLib.h"

#define ED_SIZE 0x100
```

TCP Server Client Program 예는 다음과 같다.

```
int tcpserver(int port)
{
    struct sockaddr_in servAddr;
    struct sockaddr_in clntAddr;

    int sFd;
    int cFd;
    int recvLen;
    int i;
    unsigned char data;
    int clnt_addr_size;

    if( (sFd = socket(PF_INET, SOCK_STREAM, 0)) == ERROR){
        printf("socket() Error!!\Wn");
        close(sFd);
        return ERROR;
    }
    memset(&servAddr, 0, sizeof(servAddr));
    servAddr.sin_family=AF_INET;
    servAddr.sin_addr.s_addr=htonl(INADDR_ANY);
    servAddr.sin_port=htons(port);

    if( bind(sFd, (struct sockaddr*)&servAddr, sizeof(servAddr)) != OK){
        printf("bind() Error!!\Wn");
        close(sFd);
        return ERROR;
    }
}
```



```

if( listen(sFd, 5)==ERROR){
    printf("listen() Error!!\n");
}

clnt_addr_size = sizeof(clntAddr);

if( (cFd = accept(sFd, (struct sockaddr*)&clntAddr, &clnt_addr_size)) ==
ERROR ){
    printf("accept() Error!!\n");
}

FOREVER
{
    if ( (recvLen = recv(cFd, recvBuf, sizeof(recvBuf), 0)) != ERROR){
        if(dFlag == 1)printf("\nrecvLen = %d\n", recvLen);
        for(i=0; i<recvLen; i++){
            if(!(i%16)){
                if(dFlag == 1)printf("\n");
            }

            if(dFlag == 1)printf("[%3d]",recvBuf[i]);
        }
        }else{
            printf("recv() Error!!\n");
        }
    }
}

int tcpclient(char *adrs, int port)
{
    struct sockaddr_in servAddr;
    struct sockaddr_in clntAddr;

    int i,j,k;
    int sFdC;
    int sendLen;
    unsigned char data;
    data=0;

    if( (sFdC=socket(PF_INET, SOCK_STREAM, 0))==ERROR){
        printf("socket() Error!!\n");
        close(sFdC);
        return ERROR;
    }

    memset(&servAddr, 0, sizeof(servAddr));
    servAddr.sin_family=AF_INET;
    servAddr.sin_addr.s_addr=inet_addr(adrs);
    servAddr.sin_port = port;

    for(i=0; i<sizeof(sendBuf); i++){
        sendBuf[i] = (unsigned char)data++;
    }
}

```

```

}

if( connect(sFdC, (struct sockaddr*)&servAddr, sizeof(servAddr)) == ERROR ){
    printf("connect() Error!!\n");
}

FOREVER
{
    sendLen = write(sFdC, sendBuf, sizeof(sendBuf));

    taskDelay(5);
    /*
    for(k=0; k<5000;k++)j++;
    */
    if(dFlag == 1)printf("\n send Length = %d\n", sendLen);
}
}

```

UDP Server Client Program 예는 다음과 같다.

```

int server(int port)
{
    struct sockaddr_in servAddr;
    struct sockaddr_in clntAddr;

    int sFd;
    int recvLen;
    int i;
    unsigned char data;
    int clnt_addr_size;

    if( (sFd = socket(PF_INET, SOCK_DGRAM, 0)) == ERROR){
        printf("socket() Error!!\n");
        close(sFd);
        return ERROR;
    }
    memset(&servAddr, 0, sizeof(servAddr));
    servAddr.sin_family=AF_INET;
    servAddr.sin_addr.s_addr=htonl(INADDR_ANY);
    servAddr.sin_port=htons(port);
    if( bind(sFd, (SOCKADDR*)&servAddr, sizeof(servAddr)) != OK){
        printf("bind() Error!!\n");
        close(sFd);
        return ERROR;
    }

    FOREVER
    {
        if ( (recvLen = recvfrom(sFd, recvBuf, sizeof(recvBuf), 0, (struct
sockaddr*)&clntAddr, &clnt_addr_size)) != ERROR){

```

```

        if(dFlag == 1)printf("WnrecvLen = %dWn", recvLen);
        for(i=0; i<recvLen; i++){
            if(!(i%16)){
                if(dFlag == 1)printf("Wn");
            }

            if(dFlag == 1)printf("[%3d]",recvBuf[i]);
        }
    }
}

int client(char *adrs, int port)
{
    struct sockaddr_in servAddr;
    struct sockaddr_in clntAddr;

    int i,j,k;
    int sFdC;
    int sendLen;
    unsigned char data;
    data=0;

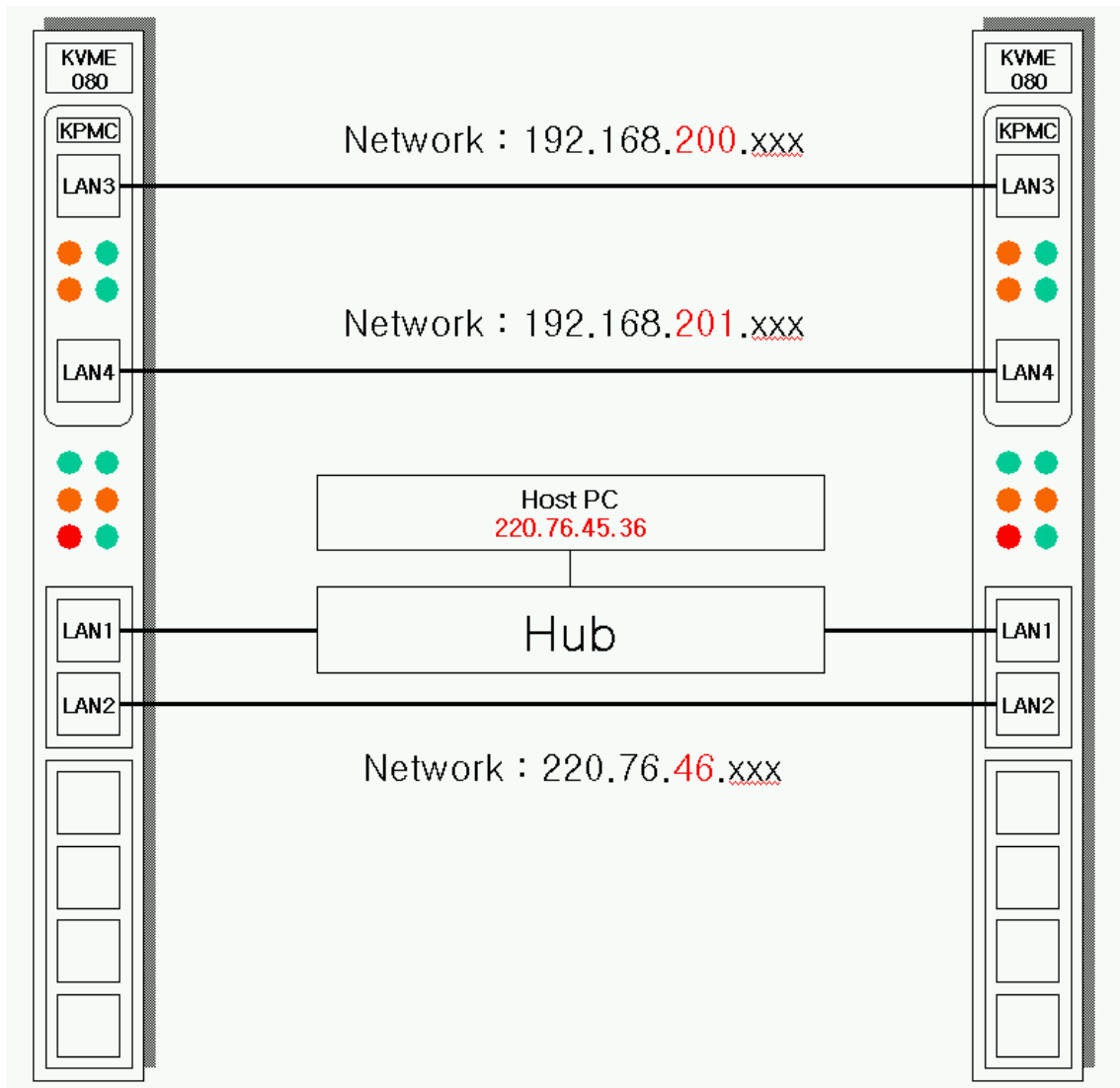
    if( (sFdC=socket(PF_INET, SOCK_DGRAM, 0))==ERROR){
        printf("socket() Error!!Wn");
        close(sFdC);
        return ERROR;
    }
    memset(&servAddr, 0, sizeof(servAddr));
    servAddr.sin_family=AF_INET;
    servAddr.sin_addr.s_addr=inet_addr(adrs);
    servAddr.sin_port = port;

    for(i=0; i<sizeof(sendBuf); i++){
        sendBuf[i] = (unsigned char)data++;
    }

    FOREVER
    {
        sendLen = sendto(sFdC, sendBuf, sizeof(sendBuf), 0, (struct
sockaddr*)&servAddr, sizeof(servAddr));
        taskDelay(1);
        /*
        for(k=0; k<5000;k++)j++;
        if(dFlag == 2)printf("Wn send Length = %dWn", sendLen);
        */
    }
}

```

TCP/UDP Program 은 위와 같다. 위의 프로그램을 이용하여, [그림 3]과 같이 환경 Setting 후 시험 할 수 있다.



[그림 3] LKP Network Test 환경

[그림 3]과 같이 구성된 Network 시험 환경에서, LKV-080A 보드를 booting 후 Network 시험 Object 를 LAN1 을 통해 다운로드 한다. 다운로드 후 Server Task 와 Client Task 를 실행하면, 각각의 LAN Port 를 TCP 통신이 된다.

080 Master 에서 실행되는 Task

```

void serverTaskA()
{
    EnetAttach(1,"220.76.46.99",0xffffffff00);
    taskDelay(10);
    EnetAttach(2,"192.168.200.100",0xffffffff00);
    taskDelay(10);
    EnetAttach(3,"192.168.201.100",0xffffffff00);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,50000);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,60000);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,70000);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,80000);
    taskDelay(10);
}

void clientTaskA()
{
    sp((FUNCPTR)tcpclient,"220.76.45.98",10000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"220.76.46.100",20000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"192.168.200.101",30000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"192.168.201.101",40000);
    taskDelay(10);
}
    
```

080 Slave 에서 실행 되는 Task

```

void serverTaskB()
{
    EnetAttach(1,"220.76.46.100",0xffffffff00);
    taskDelay(10);
    EnetAttach(2,"192.168.200.101",0xffffffff00);
    taskDelay(10);
    EnetAttach(3,"192.168.201.101",0xffffffff00);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,10000);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,20000);
    taskDelay(10);
    sp((FUNCPTR)tcpserver,30000);
}
    
```

```
taskDelay(10);
sp((FUNCPTR)tcpserver,40000);
taskDelay(10);
}

void clientTaskB()
{
    sp((FUNCPTR)tcpclient,"220.76.45.97",50000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"220.76.46.99",60000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"192.168.200.100",70000);
    taskDelay(10);
    sp((FUNCPTR)tcpclient,"192.168.201.100",80000);
    taskDelay(10);
}
```

Program 을 동작 실행하기 전 각 “LAN2”, “LAN3”, “LAN4”가 각각 Cross Cable 로 확인 후 다음과 같은 순서에 의해 실행한다.

- [순서 1] 080 master Board 에서 “serverTaskA”을 실행한다.  
“serverTaskA 를 실행하면, LKV-080A Master 의 “LAN2”, “LAN3”, “LAN4”의 IP 가 설정된다.
- [순서 2] 080 slave Board 에서 “serverTaskB”를 실행한다.  
“serverTaskB 를 실행하면, LKV-080A Slave 의 “LAN2”, “LAN3”, “LAN4”의 IP 가 설정된다.
- [순서 3] 080 master Board 에서 “clientTaskA”을 실행한다.
- [순서 4] 080 slave Board 에서 “clientTaskB”을 실행한다.

실행 순서는 위와 같으며, clientTask 에서 serverTask 로 256byte 를 연속적으로 Write 하게 된다.